

Métricas en la Administración de un Proyecto de Software

María de Lourdes Hernández Rodríguez
lourhernandez@uv.mx

Resumen. La ingeniería de software, como toda actividad humana, se encuentra en un ciclo de mejora continua. En donde la medición o aplicación de métricas es una estrategia orientada a controlar o administrar el proceso de desarrollo de software, así como dar seguimiento, evaluar y mejorar. Desde una perspectiva de alto nivel las métricas de software se focalizan en cuatro aspectos clave, a saber, (a) entender productos y procesos específicos de software (decisiones para situaciones específicas), (b) dar seguimiento, evaluar, analizar y controlar un proyecto (decisiones continuas), (c) estimar o predecir características de un producto o proceso de software y registrarlas en archivos históricos (administración basada en conocimiento), (d) Evaluar procesos o productos de software estableciendo estándares y mediciones puntuales. Este trabajo presenta un panorama general de métricas de software enfatizando el cuidado que debe ponerse en su definición y la alineación con objetivos de negocio contribuyendo así a una aplicación y análisis exitoso. En la misma línea se presentan ejemplos concretos de aplicación.

Palabras clave. Métricas, software, proceso, tipos de prueba

1 Introducción

Iniciemos este recorrido general sobre métricas de software con la popular frase “si no lo puedes medir, no lo puedes mejorar” acuñada por Lord Kelvin, un reconocido físico matemático británico. Esta frase nos invita a reflexionar en la importancia de dimensionar los elementos generados por el desarrollo o mantenimiento de software y el impacto en referencia al logro de metas.

Diversos estudios como los presentados por Jones [1], Buse y Zimmermann [2] analizan proyectos de empresas como MicroSoft e IBM, concluyendo que la mayor parte de sus proyectos utilizan métricas para toma de decisiones relativas a calidad de productos y procesos, por ejemplo registro de fallas o reportes de defectos, estimación de tiempos, entre otras; haciendo evidente la importancia de las métricas en equipos de trabajo de alto desempeño.

Algunas de las necesidades más recurrentes que cubren las métricas son: soporte a la comunicación y toma de decisiones. Considerando a los autores Pulford et al [3,4] las principales motivaciones para usar métricas pueden resumirse en:

- Planeación y estimación de proyectos
- Administración y seguimiento de proyectos

- Determinación de calidad alineada a objetivos de negocio
- Mejora de procesos, comunicación y herramientas para desarrollo de software

Los beneficios de aplicar métricas en la administración de un proyecto de software son relevantes, en particular los orientados a estimación de esfuerzo y soporte a toma de decisiones. No obstante, deben contextualizarse para impactar positivamente las actividades a realizar, de otra forma pueden representar una carga adicional de trabajo sin beneficios tangibles para la organización que implementa las métricas.

En la siguiente sección se presentan conceptos generales que permitirán al lector dar una secuencia dirigida a mostrar algunos tipos de métricas y su aplicación en contextos específicos.

1.1 Conceptos asociados a métricas

En primer término, es conveniente puntualizar el concepto asociado a métrica, “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” [6]. Como se mencionó previamente la definición de una métrica debe estar contextualizada al proyecto, producto o proceso que se atiende y debe responder a la pregunta ¿Para qué me sirve la métrica y como aplicarla? Un proceso global de aplicación es mostrado en la figura 1.

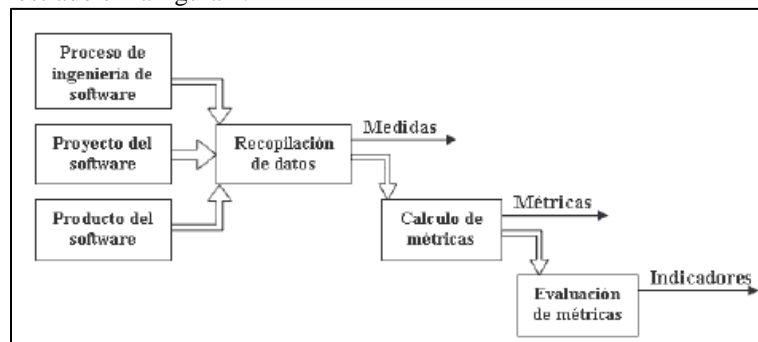


Figura 1 Proceso global de aplicación de métricas. Fuente:[5].

El concepto de métrica está asociado con una serie de términos que guían su definición y exitosa aplicación. En este sentido se presenta un meta modelo (Figura 2) que describe en una sola vista los conceptos relacionados a métricas puntualizando los elementos a considerar para la definición y documentación de una métrica, propuesta por Olsina et al [6].

Una métrica tendrá asociado un método de medición y una escala. Por medición se identifica a la actividad aplicada para obtener el valor de una medición; y medida será el número o categoría asignada a un atributo de una entidad mediante una medición [7].

Dos conceptos importantes adicionales son el método de medición, es decir, la secuencia lógica de operaciones y potenciales heurísticas aplicadas, expresada de forma genérica, y la escala: conjunto de valores con propiedades definidas, pudiendo ser nominal, ordinal, de intervalo, de razón [6].

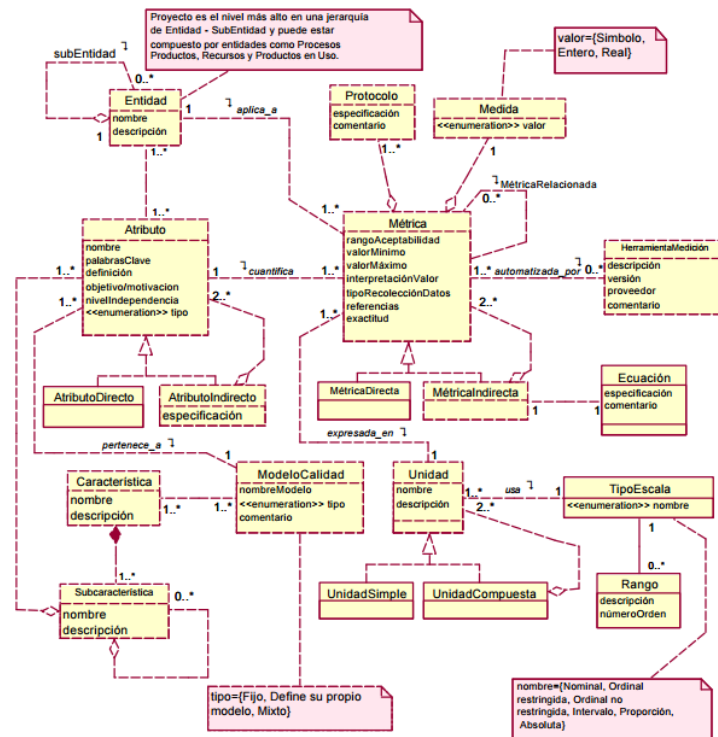


Figura 2. Meta modelo describiendo los conceptos relacionados con métricas de software a nivel proceso o producto. Fuente [6]

En referencia a las actividades asociadas a la identificación, definición, selección y aplicación de métricas o mediciones para la mejora de un proceso o proyecto de software, el estándar internacional ISO/IEC 15939 propone el proceso mostrado en la figura 3.

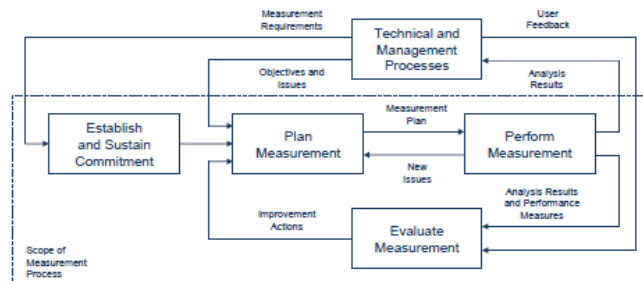


Figura 3. Proceso propuesto por el estándar ISO/IEC 15939

1.2 Tipos de métricas

La definición de métricas está estrechamente relacionada con el contexto específico para el que se requieren, el universo de definición es muy amplio por lo que pueden definirse infinidad de métricas. Desde una perspectiva práctica, se presentan las clasificaciones más comunes. La Tabla 1 presenta una clasificación alineada al proceso de desarrollo de software

Tabla 1. Métricas asociadas a modelos. Fuente: Adaptada de [8]

Elemento al que aplican	Aspectos que atienden
Modelo de análisis	Funcionalidad entregada, tamaño del sistema (funcionalmente hablando), especificación de calidad (especificidad y completitud de requerimientos), puntos de función, puntos de casos de uso
Modelos de diseño	Métricas de arquitectura (estructura, complejidad), métricas a nivel de componentes (complejidad), métricas de interfaz de usuario (grado de usabilidad), métricas a nivel de clases (interfaces, acoplamiento, jerarquía, colaboración)
Código fuente	Métricas de complejidad (cohesión, acoplamiento, complejidad ciclomática), métricas de longitud (líneas de código, número de entradas, número de salidas)
A nivel de pruebas	Métricas de cobertura funcional, métricas a nivel de casos de prueba, métricas a nivel unidad, integración o sistemas, número de defectos permitidos por tipo.

Ahora bien, desde la perspectiva de control de un proyecto de software las dimensiones en las que se visualizan métricas se muestran en la Figura 4. En donde destacan los elementos clave: Personas, Tecnología y Producto asociados a Procesos.

En este caso la eficacia de un proceso de software se mide de forma indirecta. Esto a través de un conjunto de métricas derivadas de los productos generados al realizar un proceso de desarrollo de software. Incluyendo: defectos encontrados y reportados por usuarios finales, tiempo de incorporación al proceso productivo, tiempo invertido en aprender a operar el nuevo sistema, índice de productividad alcanzado al adoptar el software, entre otras.

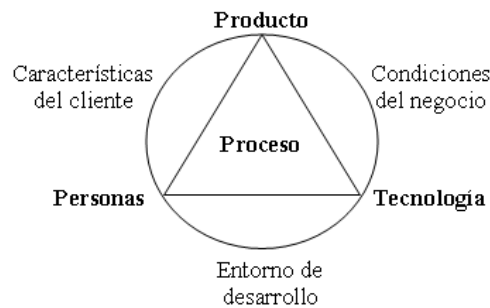


Figura 4. Aspectos determinantes de la calidad en software. Fuente [8]

1.3 Técnicas para definición de un programa de métricas

Existen diferentes propuestas para establecer un programa de métricas en una organización. Dos propuestas que tienen asociados casos de éxito son las propuestas por el Software Engineering Institute [9] orientada por metas, y la propuesta por el Software Productivity Center [5] orientada por el tipo de información o datos, procesos y control de calidad aplicado. A manera de resumen a continuación se mencionan brevemente algunas técnicas o estrategias presentadas por Pressman [5].

Goal-Question-Metrics: técnica definida por Basili y Weiss para seleccionar y generar métricas tanto del proceso como de los resultados de un proyecto. Conformado por tres niveles: conceptual (goals), operacional (question), cuantitativo (metrics) [9].

GQ(I)M Metodología que identifica y define métricas de software que dan soporte a objetivos de negocio de la empresa a nivel de procesos y proyectos. Define el quien, que, cuando, porque y como de los indicadores.

PSM: Practical Software Measurement Basado en experiencias obtenidas por las organizaciones para saber la mejor forma de implementar un programa de medición de software con éxito.

Modelos de métricas de software: ISO 9000 (ISO 9000, ISO 9001, ISO 9003, ISO 9000-3) ISO/IEC 9126. Considerando los atributos: funcionalidad(adaptabilidad, exactitud, interoperabilidad, seguridad), confiabilidad (madurez, tolerancia a fallas, recuperabilidad), usabilidad (comprensibilidad, aprendizaje, operabilidad), eficiencia(comportamiento en función a tiempo, uso de los recursos), mantenimiento (análisis, cambio, estabilidad, prueba).

QSOS Method for qualification and selection of open source software. Propone la evaluación, cuantificación y selección de software open source.

2 Ejemplos de aplicación de métricas

Una vez introducidos los temas de conceptos y estrategias de definición de métricas, seguramente el lector observará que realizar un programa de aplicación de métricas no es trivial. Pero si tenemos la contextualización suficiente en el dominio de negocio y el conocimiento sobre métricas podemos trabajar al respecto con objetivos discretos y gradualmente alcanzar metas mayores de control.

A manera de ejemplo la Tabla 2 presenta algunos aspectos de interés en el control de un proyecto y métricas que podrían aplicarse.

Tabla 2. Ejemplos concretos de aplicación de métricas

Aspecto	Atributos	Métricas
Proceso de desarrollo	Tiempo invertido	Días calendario, días trabajados
	Puntos de control	Fechas específicas en calendario
	Fase de depuración	Porcentaje total de defectos encontrados
	Fase de entrega o cierre	Porcentaje de tareas completas
Proceso de pruebas	Volumen	Número de pruebas programadas
	Progreso	Número de pruebas ejecutadas Número de pruebas pasadas
Mantenimiento	Costo	Inversión monetaria por año Horas invertidas por cambio
Sistema	Tamaño	Número de módulos Puntos de función Puntos de casos de uso Memoria requerida
Grupo de trabajo asignado	Tamaño	Número de personal asignado
	Experiencia	Años de experiencia en el dominio de negocio Años de experiencia en programación

Ahora bien, si tomamos un ejemplo concreto aplicando GQM, dado que es una estrategia relativamente simple y completa, podría analizarse el siguiente escenario.

Asuma que es el administrador de un equipo de desarrollo de software y debe decidir la fecha de liberación del sistema que está desarrollando. Aplicando GQM construiría un árbol de decisión como el de la figura 5.

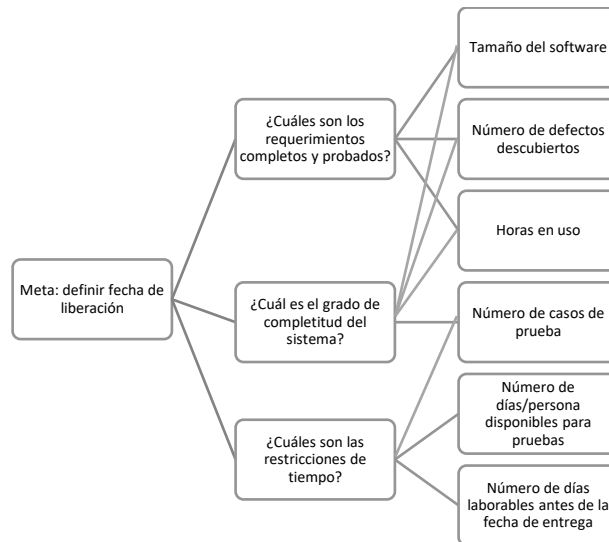


Figura 5 Ejemplo de definición de métricas aplicando GQM

A través de esta estrategia se establece el objetivo o meta y se asocian al mismo preguntas que ayudarán a determinar información asociada a las misma, la respuesta a las preguntas planteadas determinará la métricas que deben aplicarse.

El resto del proceso consiste en un análisis de las métricas su asociación a indicadores y en función a estos la toma de decisiones.

3 Conclusiones sobre métricas

Indiscutiblemente las métricas juegan un rol muy importante en la administración de un proceso de desarrollo de software soportando la toma de decisiones, la predicción o estimación, la gestión y el control. Sin embargo, existen situaciones que deben atenderse para asegurar un éxito en su análisis y aplicación como las siguientes:

- Demostrar la utilidad de una métrica a todos los involucrados en un proceso de desarrollo o mantenimiento de software, considerando el dominio de negocio que se atiende.
- Focalizarse en la medición de un conjunto acotado de atributos de calidad que se complementen y permitan una mejor explicación del impacto que tienen en un producto de software
- Generalmente se tienen tiempos limitados para recolección de datos e interpretación de una métrica, de ahí la importancia de utilizar herramientas de soporte.
- Una métrica debe definirse en función a los objetivos de negocio que apoya y su interpretación debe ser cuidadosa y objetiva. La recolección de información y la obtención de métricas solo es una parte del trabajo.

Como se mencionó al principio la toma de decisiones soportada en la experiencia previa ayudará a definir e interpretar las métricas más eficientemente.

Referencias

1. C. Jones, *Applied Software Measurement: Global Analysis of Productivity and Quality*, vol. 3, McGraw-hill, New York, 2008.
2. Raymond PL Buse, Thomas Zimmermann, Information needs for software development analytics, in: *Proceedings of the 2012 International Conference on Software Engineering*, IEEE Press, 2012, pp. 987–996.
3. K. Pulford, A. Kuntzmann-Combelles, S. Shirlaw, *A Quantitative Approach to Software Management: the AMI Handbook*, Addison-Wesley Longman Publishing Co., Inc., 1995.
4. R.B. Grady, *Practical Software Metrics for Project Management and Process Improvement*, Prentice-Hall, Inc., 1992.
5. R. S. Pressman. *Ingeniería de Software: un enfoque práctico*. 5ta ed. Ed. McGraw-Hill. México. 2001. pp. 53-75
6. Olsina, L., Bertoa, M. F., Lafuente, G., Martín, M. A., Matrib, M., & Vallecillo, A. (2002, November). Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad. In *JISBD* (pp. 189-200).
7. Radatz, J., Geraci, A., & Katki, F. (1990). IEEE standard glossary of software engineering terminology. *IEEE Std, 610121990(121990)*, 3.
8. Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Sixth Edition Palgrave Macmillan.
9. Park, R. E., Goethert, W. B., & Florac, W. A. (1996). *Goal-Driven Software Measurement. A Guidebook* (No. CMU/SEI-96-HB-002). Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.